

Fig. 1

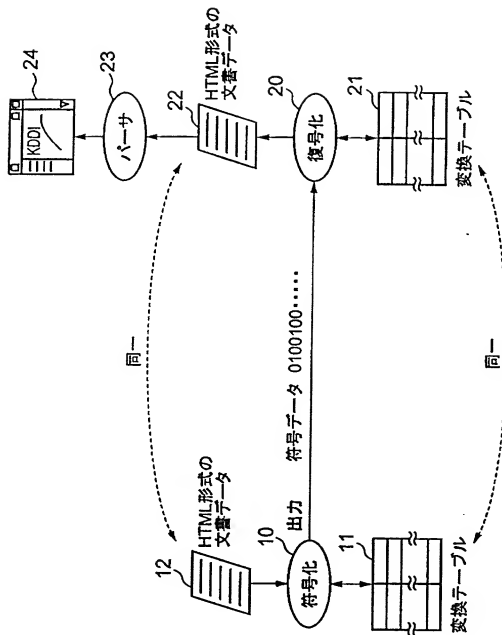


Fig. 2

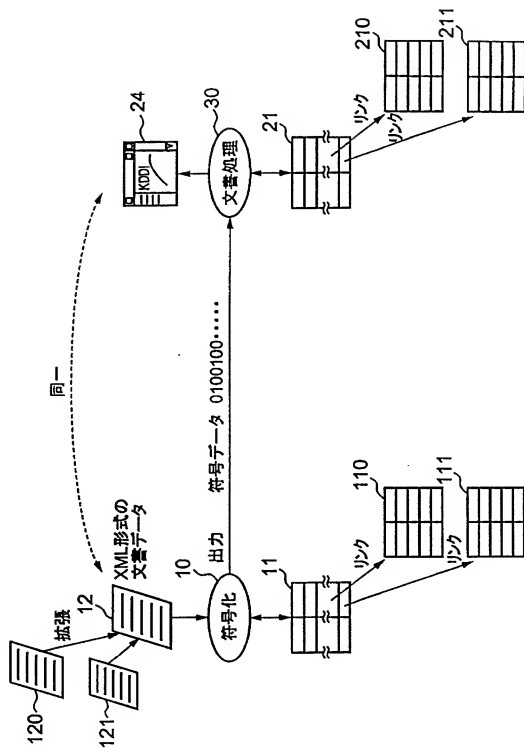


Fig. 3

[文書データのサンプル]

```
<svg width="500" height="800" > [1]
  <rect x="0" y="0" width="10" height="100" /> [2]
  <text x="480" y="300">文字列 1</text> [3]
</svg> [4]
```

10025610.122601

Fig. 4

[符号データのサンプル]

00 000	名"00"+要素 SVG の開始"000" [1]
00 10	名"00"+属性名 width"10"
01 0111110100	数値"01"+属性値 500
00 11	名"00"+属性名 height"11"
01 1100100000	数値"01"+属性値 500
00 001	名"00"+要素 rect の開始 [2]
00 100	名"00"+属性名 x"100"
01 0000000000	数値"01"+属性値 0
00 101	名"00"+属性名 y"101"
01 0000000000	数値"01"+属性値 0
00 110	名"00"+属性名 width"110"
01 0000001010	数値"01"+属性値 10
00 111	名"00"+属性名 height"111"
01 0001100100	数値"01"+属性値 100
00 011	名"00"+要素 rect の終了"011"
00 010	名"00"+要素 text の開始"010" [3]
00 10	名"00"+属性名 x"10"
01 0111100000	数値"01"+属性値 480
00 11	名"00"+属性名 y"11"
01 0100101100	数値"01"+属性値 300
10 00000100	文字列"10"+文字数 4
1001010110110110	文字「文」
1000111010011010	文字「字」
1001011111110001	文字「列」
1000001001010000	文字「1」
00 011	名"00"+要素 text の終了"011"
00 011	名"00"+要素 svg の終了"011" [4]

10025670.122601

Fig. 5a

[変換テーブルのサンプル]

```
<?xml version="1.0"?>
<xeus version="1.0" >
<head>
  <prefix bit="2">
    <name_prefix code="00" />
    <number_prefix code="01" />
    <char_prefix code="10" />
  </prefix>
  <root name="svg" bit="3" code="000" />
  <end name="/svg" bit="3" code="011" />
</head>
```

((1)ヘッダ部分の開始)

((2)符号長)

((3)名を示す符号)

((4)数値を示す符号)

((5)文字列を示す符号)

((6)要素 SVG の開始に 3bit"000"を割当)

((7)要素 SVG の終了に 3bit"011"を割当)

Fig. 5b

```

<body>                                ((8) ボディ部分の開始)
<element name="svg">                    ((9) 要素名 svg を表す)
  <attlist bit="2">                      ((10) 属性名に 2bit 割当)
    <attr name="width" code="10">
      ((11) 要素名 svg に付随する属性名 width に "10" 割当)
      <value>
        <number bit="10" data="UI" qt="1" />
          ((12) 属性名 width の属性値は符号無し整数 10bit で表す)
      </value>
    </attr>
    <attr name="height" code="11">
      ((13) 要素名 svg に付随する属性名 height に "11" 割当)
      <value>
        <number bit="10" data="UI" qt="1" />
          ((14) 属性名 height の属性値は符号無し整数 10bit で表す)
      </value>
    </attr>
  </attlist>
  <children bit="3">                    ((15) 子要素を 3bit で表す)
    <child_element name="rect" code="001" end-code="011"/>
      ((16) 要素名 svg の子関係にある要素名 rect を表す)
      ((17) 要素名 rect の開始に "001" を割当、終了に "011" 割当)
    <child_element name="text" code="010" end-code="011"/>
      ((18) 要素名 svg の子関係にある要素名 text を表す)
      ((19) 要素名 text の開始に "010" を割当、終了に "011" 割当)
  </children>
</element>

```

Fig. 5c

```

<element name="rect">          ((20)要素名 rect を表す)
  <attlist bit="3">           ((21)属性名に 3bit 割当)
    <attr name="x" code="100">  ((22)属性名 x に "100" を割当)
      <value>
        <number bit="10" data="I" qt="implied" />
          ((23)属性名 x の属性値は、符号付き整数 10bit で表す)
      </value>
    </attr>
    <attr name="y" code="101">  ((24)属性名 y に "101" を割当)
      <value>
        <number bit="10" data="I" qt="implied" />
          ((25)属性名 y の属性値は符号付き整数 10bit で表す)
      </value>
    </attr>
    <attr name="width" code="110">
      ((26)属性名 width に "110" を割当)
      <value>
        <number bit="10" data="UI" qt="implied" />
          ((27)属性名 width の属性値は符号無し整数 10bit で表す)
      </value>
    </attr>
    <attr name="height" code="111">
      ((28)属性名 height に "111" を割当)
      <value>
        <number bit="10" data="UI" qt="implied" />
          ((29)属性名 height の属性値は符号無し整数 10bit で表す)
      </value>
    </attr>
  </attlist>
</element>

```

Fig. 5d

```

<element name="text">          ((30)要素名 textを表す)
  <attlist bit="2">            ((31)属性名に 2bit 割当)
    <attr name="x" code="10">    ((32)属性名 x に "10"割当)
      <value>
        <number bit="10" data="I" qt="implied" />
          ((33)属性名 x の属性値は符号付き整数 10bit で表す)
      </value>
    </attr>
    <attr name="y" code="11">    ((34)属性名 x に "11"割当)
      <value>
        <number bit="10" data="I" qt="implied" />
          ((35)属性名 y の属性値は符号付き整数 10bit で表す)
      </value>
    </attr>
  </attlist>
  <element_value>              ((36)要素 text の要素値を表す)
    <char length="implied" encoding="Shift_JIS" qt="1" />
      ((37)要素値が Shift_JIS 形式であることを表す)
  </element_value>
</element>
</body>
</xex>

```



Fig. 6

文書 A

```
<a1>
  <a2>aaa</a2>
</a1>
```

文書 B

```
<b1>bbb</b1>
```

文書 A と文書 B との混合

```
<a:a1 xmlns:a="#a" xmlns:b="#b">
  <b:b1>bbb</b:b1>
  <a:a2>aaa</a:a2>
</a:a1>
```

変換テーブル

```
<?xml version="1.0"?>
<xeus xmlns="xeus-for-a&b">
<head>
  <namespaces>
    <namespace bit="1" name="a1" prefix="a" code="0"
                                     xlink:href="a.xeus"/>
    <namespace bit="1" name="a1" prefix="b" code="1"
                                     xlink:href="b.xeus"/>
  </namespaces>
</head>
</xeus>
```

Fig. 7

[符号データのサンプル]

00 0000000100010010 000	名"00"+占有 274bit+要素 SVG の開始"000" [1]
00 10	名"00"+属性名 width"10"
01 0111110100	数値"01"+属性値 500
00 11	名"00"+属性名 height"11"
01 1100100000	数値"01"+属性値 500
00 0000000001001100 001	名"00"+占有 76bit+要素 rect の開始 [2]
00 100	名"00"+属性名 x"100"
01 0000000000	数値"01"+属性値 0
00 101	名"00"+属性名 y"101"
01 0000000000	数値"01"+属性値 0
00 110	名"00"+属性名 width"110"
01 0000001010	数値"01"+属性値 10
00 111	名"00"+属性名 height"111"
01 0001100100	数値"01"+属性値 100
00 011	名"00"+要素名 rect の終了"011"
00 0000000001110010 010	名"00"+占有 114bit+要素 text の開始"010" [3]
00 10	名"00"+属性名 x"10"
01 0111100000	数値"01"+属性値 480
00 11	名"00"+属性名 y"11"
01 0100101100	数値"01"+属性値 300
10 00000100	文字列"10"+文字数 4
1001010110110110	文字「文」
1000111010011010	文字「字」
1001011111110001	文字「列」
1000001001010000	文字「1」
00 011	名"00"+要素 text の終了"011"
00 011	名"00"+要素 svg の終了"011" [4]

Fig. 8

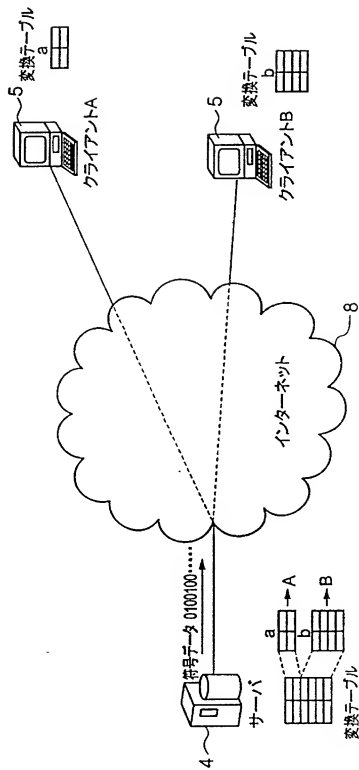


Fig. 9

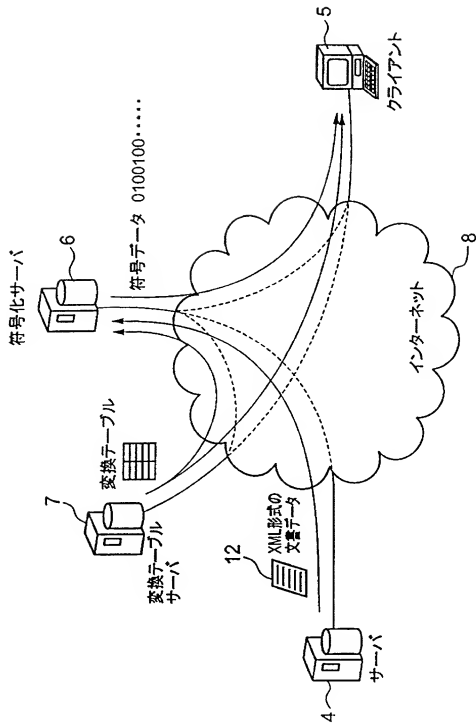
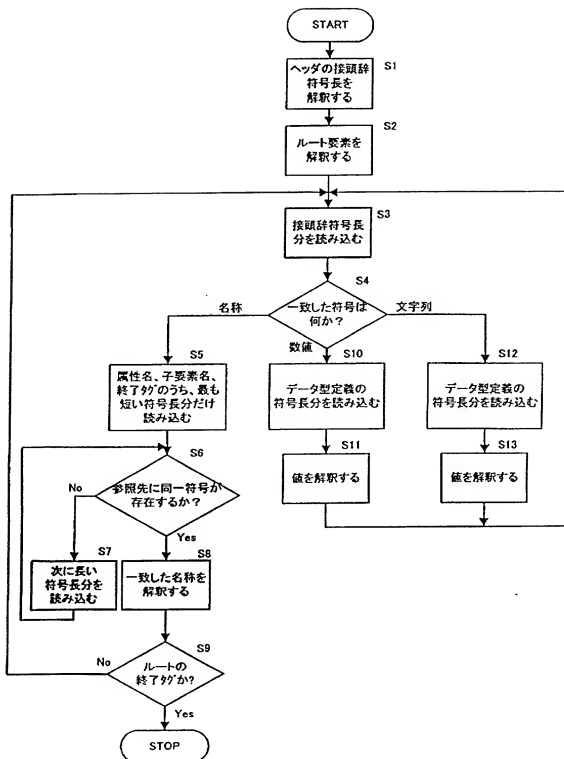


Fig. 10



10025610.122601